

CLAIMS

What is claimed is:

- 1 1. A method of a development and build environment for packaged software delivery in
2 a distributed network of nodes, the method comprising the computer-implemented steps of:
3 compiling source code files into executable file modules;
4 wherein a module contains an image for a process or a dynamically linked library
5 (DLL);
6 creating a software package that comprises at least one module;
7 wherein packages are created based on features/characteristics or purpose;
8 creating metadata for a module that includes, but is not limited to, information such as
9 the module's: binary signature, name, directory path, and characteristics;
10 inserting the module's metadata into the software package;
11 gathering application program interface (API) dependency information for each
12 module;
13 wherein a module can provide and use at least one API;
14 collecting dependencies documented in module specifications and placing them into
15 the module's metadata; and
16 wherein the dependencies documented in each module lists API names and versions
17 that the process or DLL depends on.
- 1 2. A method as recited in Claim 1, further comprising the step of:
2 providing a linker; and
3 wherein said linker creates a list of dependent modules for a given process or DLL
4 module and places the list in the module's metadata.
- 1 3. A method as recited in Claim 1, further comprising the step of:
2 creating metadata for each API;
3 inserting the API metadata into the software package; and

4 wherein metadata for an API includes, but is not limited to: the API's name and
5 version.

1 4. A method as recited in Claim 1, further comprising the step of:
2 calculating a binary signature for each module and inserting the binary signature into
3 the respective module's metadata; and
4 wherein each unique version of a module will have a unique binary signature.

1 5. A method as recited in Claim 1, further comprising the step of:
2 creating metadata for a package that includes, but is not limited to, information such
3 as the package's: name, build date, and characteristics; and
4 inserting the package metadata into the package.

1 6. A method of a development and build environment for packaged software delivery in
2 a distributed network of nodes, the method comprising the computer-implemented steps of:
3 compiling source code files into executable file modules;
4 wherein a module contains an image for a process or a dynamically linked library
5 (DLL);
6 creating a software package that comprises at least one module;
7 creating metadata for a module that includes, but is not limited to, information such as
8 the module's: binary signature, name, directory path, and characteristics;
9 inserting the module's metadata into the software package;
10 gathering application program interface (API) dependency information for each
11 module; and
12 wherein a module can provide and use at least one API.

1 7. A method as recited in Claim 6, further comprising the step of:
2 providing a linker; and
3 wherein said linker creates a list of dependent modules for a given process or DLL
4 module and places the list in the module's metadata.

- 1 8. A method as recited in Claim 6, further comprising the step of:
2 collecting dependencies documented in module specifications and placing them into
3 the module's metadata; and
4 wherein the dependencies documented in each module lists API names and versions
5 that the process or DLL depends on.
- 1 9. A method as recited in Claim 6, further comprising the step of:
2 creating metadata for each API;
3 inserting the API metadata into the software package; and
4 wherein metadata for an API includes, but is not limited to: the API's name and
5 version.
- 1 10. A method as recited in Claim 6, further comprising the step of:
2 calculating a binary signature for each module and inserting the binary signature into
3 the respective module's metadata; and
4 wherein each unique version of a module will have a unique binary signature.
- 1 11. A method as recited in Claim 6, wherein packages are created based on
2 features/characteristics or purpose.
- 1 12. A method as recited in Claim 6, further comprising the step of:
2 creating metadata for a package that includes, but is not limited to, information such
3 as the package's: name, build date, and characteristics; and
4 inserting the package metadata into the package.
- 1 13. An apparatus for a development and build environment for packaged software
2 delivery in a distributed network of nodes, comprising:
3 means for compiling source code files into executable file modules;

wherein a module contains an image for a process or a dynamically linked library (DLL);
means for creating a software package that comprises at least one module;
means for creating metadata for a module that includes, but is not limited to, information such as the module's: binary signature, name, directory path, and characteristics;
means for inserting the module's metadata into the software package;
means for gathering application program interface (API) dependency information for each module; and
wherein a module can provide and use at least one API.

14. An apparatus as recited in Claim 13, further comprising the step of:
a linker; and
wherein said linker creates a list of dependent modules for a given process or DLL module and places the list in the module's metadata.

15. An apparatus as recited in Claim 13, further comprising the step of:
means for collecting dependencies documented in module specifications and placing them into the module's metadata; and
wherein the dependencies documented in each module lists API names and versions that the process or DLL depends on.

16. An apparatus as recited in Claim 13, further comprising the step of:
means for creating metadata for each API;
means for inserting the API metadata into the software package; and
wherein metadata for an API includes, but is not limited to: the API's name and version.

17. An apparatus as recited in Claim 13, further comprising the step of:
means for calculating a binary signature for each module and inserting the binary signature into the respective module's metadata; and

4 wherein each unique version of a module will have a unique binary signature.

1 18. An apparatus as recited in Claim 13, wherein packages are created based on
2 features/characteristics or purpose.

1 19. An apparatus as recited in Claim 13, further comprising the step of:
2 means for creating metadata for a package that includes, but is not limited to,
3 information such as the package's: name, build date, and characteristics; and
4 means for inserting the package metadata into the package.

1 20. A computer-readable medium carrying one or more sequences of instructions for a
2 development and build environment for packaged software delivery in a distributed network
3 of nodes, which instructions, when executed by one or more processors, cause the one or
4 more processors to carry out the steps of:
5 compiling source code files into executable file modules;
6 wherein a module contains an image for a process or a dynamically linked library
7 (DLL);
8 creating a software package that comprises at least one module;
9 creating metadata for a module that includes, but is not limited to, information such as
10 the module's: binary signature, name, directory path, and characteristics;
11 inserting the module's metadata into the software package;
12 gathering application program interface (API) dependency information for each
13 module; and
14 wherein a module can provide and use at least one API.

1 21. A computer-readable medium as recited in Claim 20, further comprising the step of:
2 providing a linker; and
3 wherein said linker creates a list of dependent modules for a given process or DLL
4 module and places the list in the module's metadata.

1 22. A computer-readable medium as recited in Claim 20, further comprising the step of:
2 collecting dependencies documented in module specifications and placing them into
3 the module's metadata; and
4 wherein the dependencies documented in each module lists API names and versions
5 that the process or DLL depends on.

1 23. A computer-readable medium as recited in Claim 20, further comprising the step of:
2 creating metadata for each API;
3 inserting the API metadata into the software package; and
4 wherein metadata for an API includes, but is not limited to: the API's name and
5 version.

1 24. A computer-readable medium as recited in Claim 20, further comprising the step of:
2 calculating a binary signature for each module and inserting the binary signature into
3 the respective module's metadata; and
4 wherein each unique version of a module will have a unique binary signature.

1 25. A computer-readable medium as recited in Claim 20, wherein packages are created
2 based on features/characteristics or purpose.

1 26. A computer-readable medium as recited in Claim 20, further comprising the step of:
2 creating metadata for a package that includes, but is not limited to, information such
3 as the package's: name, build date, and characteristics; and
4 inserting the package metadata into the package.